

Fig. 7

Send, from TIU_0 to GC_0 and GC_T , a setup message for the call ~210

Establish a gate at NED_T upon receiving the setup message from GC_T ~220

Establish a gate at NED_0 upon receiving the setup message from GC_0 ~230

Send a reserve message from TIU_0 to NED_0 ~240

Send a reserve message from TIU_T to NED_T ~250

Exchange end-to-end message between TIU_0 and TIU_T ~260

Upon connecting the calling party and the called party, send a commit message from TIU_0 to NED_0 and from TIU_T to NED_T ~270

Upon receiving the commit message at NED_0 , open the gate at NED_0 ~280

Upon receiving the commit message at NED_T , open the gate at NED_T ~290

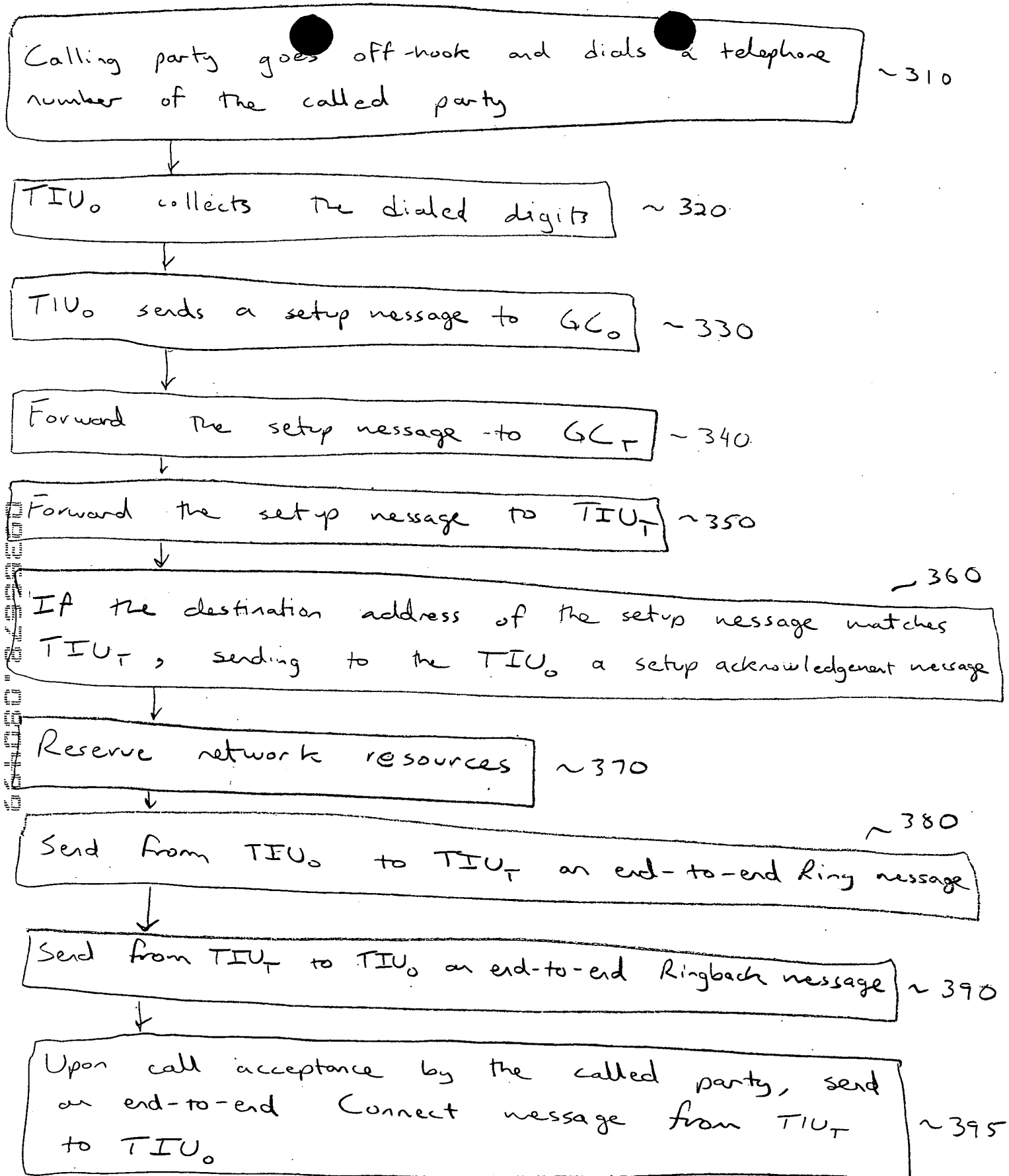


FIG. 3

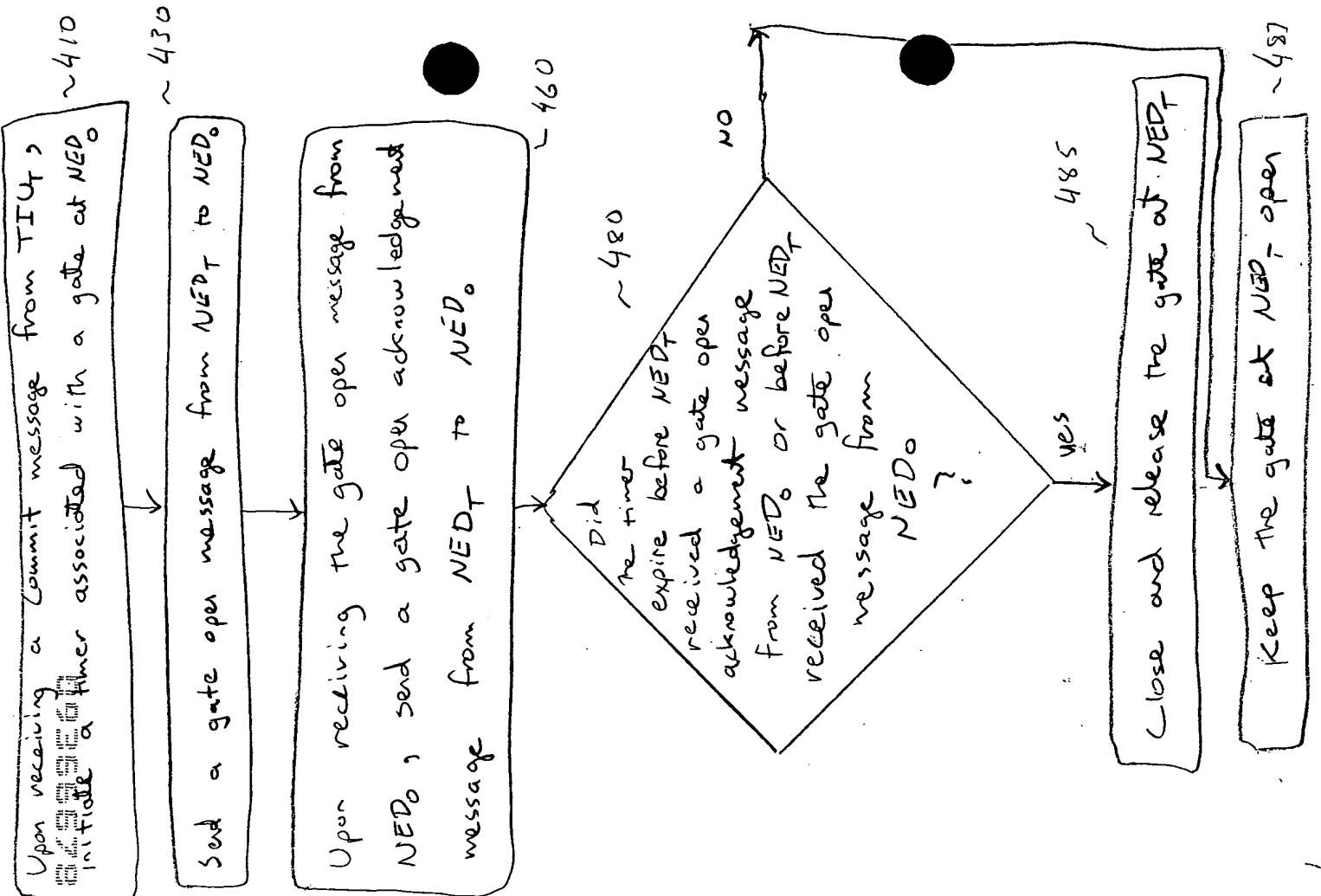
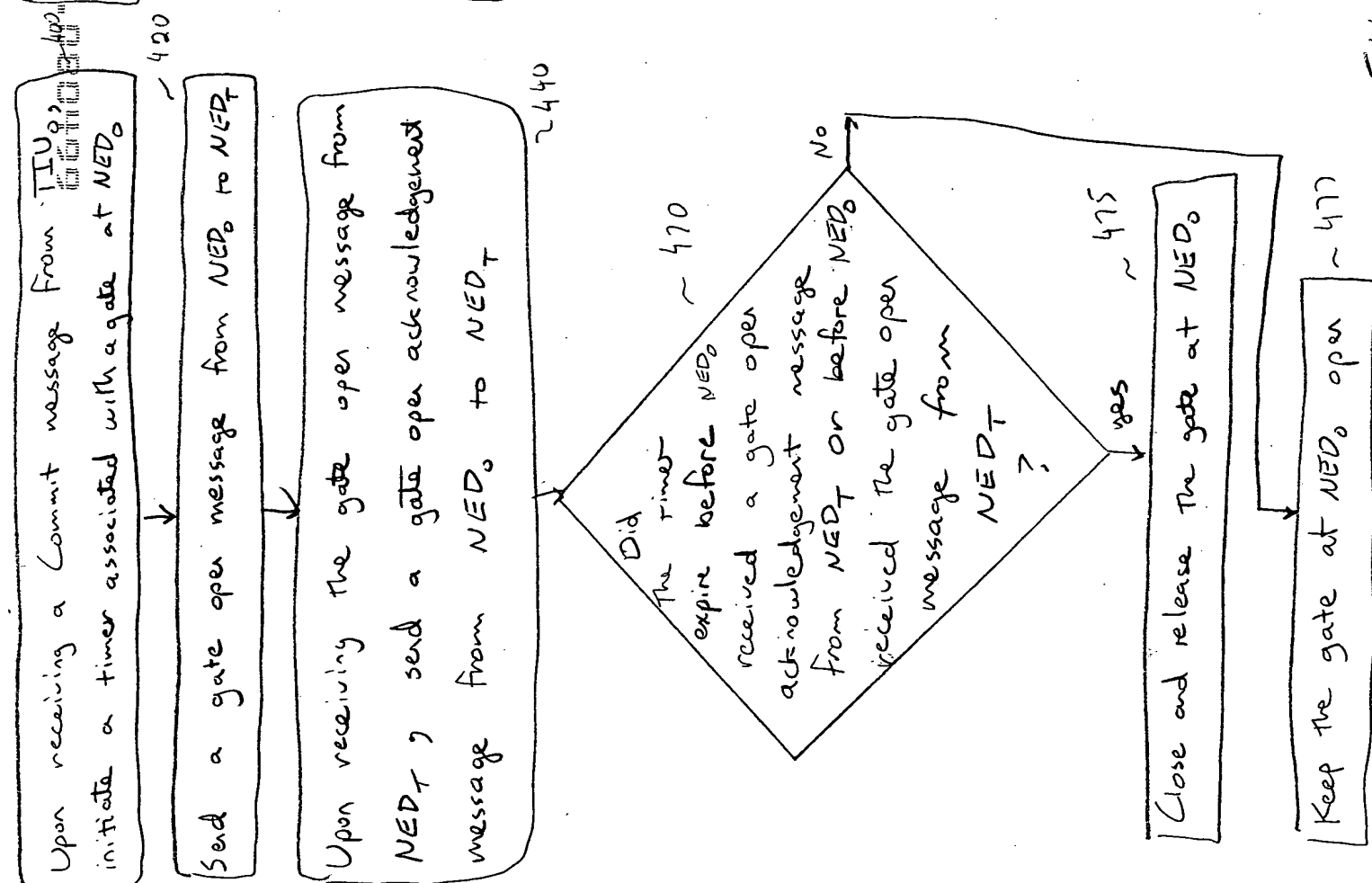


FIG. 4

Packets are sent from TIU₀ to NED₀ ~ 500

Translate the local source address and local destination address to a global source address and a global destination address ~510

Forward the translated packets from NED_0 to NED_T

Translate the global source address and the global destination address to a second local source address, and a second local address ~530

Send the translated packets from NED_T to TIU_T ~ 590

Flg. 5

```

sequenceDiagram
    participant BTI_O as BTI_O
    participant ER_O as ER_O
    participant GC_O as GC_O
    participant GC_T as GC_T
    participant ER_T as ER_T
    participant BTI_T as BTI_T

    Note over BTI_O: Provides dialtone.  
Collects complete E.164_T.
    BTI_O->>ER_O: SETUP
    Note over ER_O,GC_O: Authenticates BTI_O via AI. Matches  
E.164_T to GC_T. Determines there is  
sufficient network capacity for call.
    GC_O->>ER_O: GATEALLOC
    Note over ER_O: Allocates the  
gate for this call.
    ER_O->>GC_O: GATEALLOCACK
    GC_O->>GC_T: GCSETUP
    Note over GC_T: Translates E.164_T to BTI_T and  
matches with ER_T.
    GC_T->>ER_T: GATESETUP
    Note over ER_T: "Establishes the gate" for this call.
    ER_T->>GC_T: GATESETUPACK
    GC_T->>BTI_T: SETUP
    BTI_T->>GC_T: SETUPACK
    GC_T->>GC_O: GCSETUPACK
    GC_O->>ER_O: GATESETUP
    Note over ER_O: "Establishes the gate" for this call.
    ER_O->>BTI_O: GATESETUPACK
    BTI_O->>GC_O: SETUPACK
    Note over BTI_O,ER_O,GC_O,GC_T,ER_T,BTI_T: Additional Capability Negotiation (if necessary)
    Note over BTI_O,ER_O,GC_O,GC_T,ER_T,BTI_T: Intermediate Routers
    Note over BTI_O,ER_O,GC_O,GC_T,ER_T,BTI_T: Reserves Backbone Channel
    BTI_O->>ER_O: RESERVE
    ER_O->>BTI_O: RESERVEACK
    Note over ER_O,GC_O: Reserves access channel
    GC_O->>ER_O: RESERVE
    ER_O->>GC_O: RESERVEACK
    Note over ER_O,GC_T: Reserves access channel
    GC_T->>ER_T: RESERVE
    ER_T->>GC_T: RESERVEACK
    Note over BTI_O,ER_O,GC_O,GC_T,ER_T,BTI_T: RING
    Note over BTI_O,ER_O,GC_O,GC_T,ER_T,BTI_T: RINGBACK
    Note over BTI_O: Starts ringback
    Note over ER_T,BTI_T: Rings phone
    Note over ER_T,BTI_T: Detects off-hook
    BTI_O->>ER_O: CONNECT
    ER_O->>BTI_T: COMMIT
    BTI_T->>ER_T: COMMIT
    Note over ER_O,GC_O: Allocates access channel.
    GC_O->>ER_O: GATEOPEN
    Note over ER_O: Generates call record.
    ER_O->>GC_T: GATEOPEN
    Note over GC_T: Allocates access channel.
    GC_T->>ER_T: GATEOPEN
    Note over ER_T: Generates call record.
    ER_T->>GC_O: GATEOPENACK
    GC_O->>ER_O: GATEOPENACK
    ER_O->>BTI_O: COMMITACK
    BTI_O->>GC_O: COMMITACK
    GC_T->>ER_T: COMMITACK
    ER_T->>GC_T: COMMITACK
    Note over BTI_O,ER_O,GC_O,GC_T,ER_T,BTI_T: Call In Progress
  
```

Figure 6

```

sequenceDiagram
    participant BTI_O as BTIO
    participant ER_O as ERO
    participant BR1 as Backbone Router
    participant BR2 as Backbone Router
    participant ER_T as ERT
    participant BTI_T as BTIT

    BTI_O->>ER_O: RESERVE
    Note over ER_O: Checks that GIDO is set up. Looks up GAO and GAT. Checks that upstream capacity is available in access and reserves. Reserves capacity in backbone network.
    ER_O->>BR1: BACKBONERESERVE
    Note over BR1: Checks that forward capacity is available and reserves. Forwards reservation message.
    BR1->>BR2: BACKBONERESERVE
    Note over BR2: Checks that forward capacity is available and reserves. Forwards reservation message.
    BR2->>ER_T: BACKBONERESERVE
    ER_T->>BTI_T: RESERVE
    BTI_T->>ER_T: RESERVE
    ER_T->>BR2: BACKBONERESERVE
    BR2->>BR1: BACKBONERESERVE
    BR1->>ER_O: BACKBONERESERVE
    ER_O->>BTI_O: RESERVE-ACK (TIDO)
    BTI_O->>BTI_T: BACKBONERESERVEACK
    BTI_T->>ER_T: RESERVEACK
    ER_T->>BR2: BACKBONERESERVE
    Note over ER_T: Matches GAO, PNO, GAT, PNT with gate GIDT.
  
```

The diagram illustrates the reservation process across a multi-hop network. It involves five entities: BTI_O , ER_O , two **Backbone Router** nodes, ER_T , and BTI_T . The process begins with BTI_O sending a **RESERVE** message to ER_O . ER_O then sends **BACKBONERESERVE** to the first Backbone Router, which forwards it to the second Backbone Router, and finally to ER_T . ER_T sends a **RESERVE** message to BTI_T , which also sends a **RESERVE** message back to ER_T . ER_T then sends **BACKBONERESERVE** back to the second Backbone Router, which forwards it to the first Backbone Router, and finally to ER_O . ER_O sends a **RESERVE-ACK (TID_O)** message to BTI_O . BTI_O then sends a **BACKBONERESERVEACK** message to BTI_T , which sends a **RESERVEACK** message to ER_T . Finally, ER_T sends a **BACKBONERESERVE** message to the second Backbone Router, which forwards it to the first Backbone Router, and finally to ER_O . A note indicates that ER_T matches GA_O , PN_O , GA_T , PN_T with gate GID_T .

Figure 7

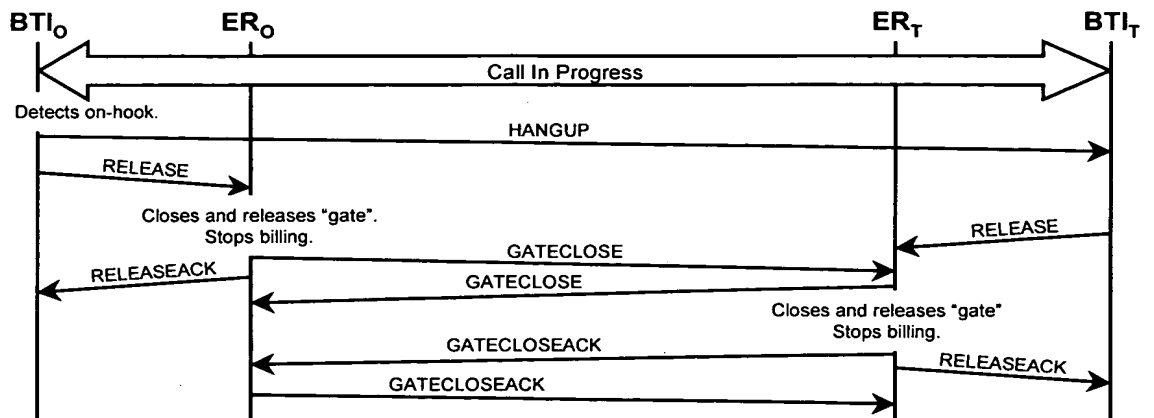


Figure 8

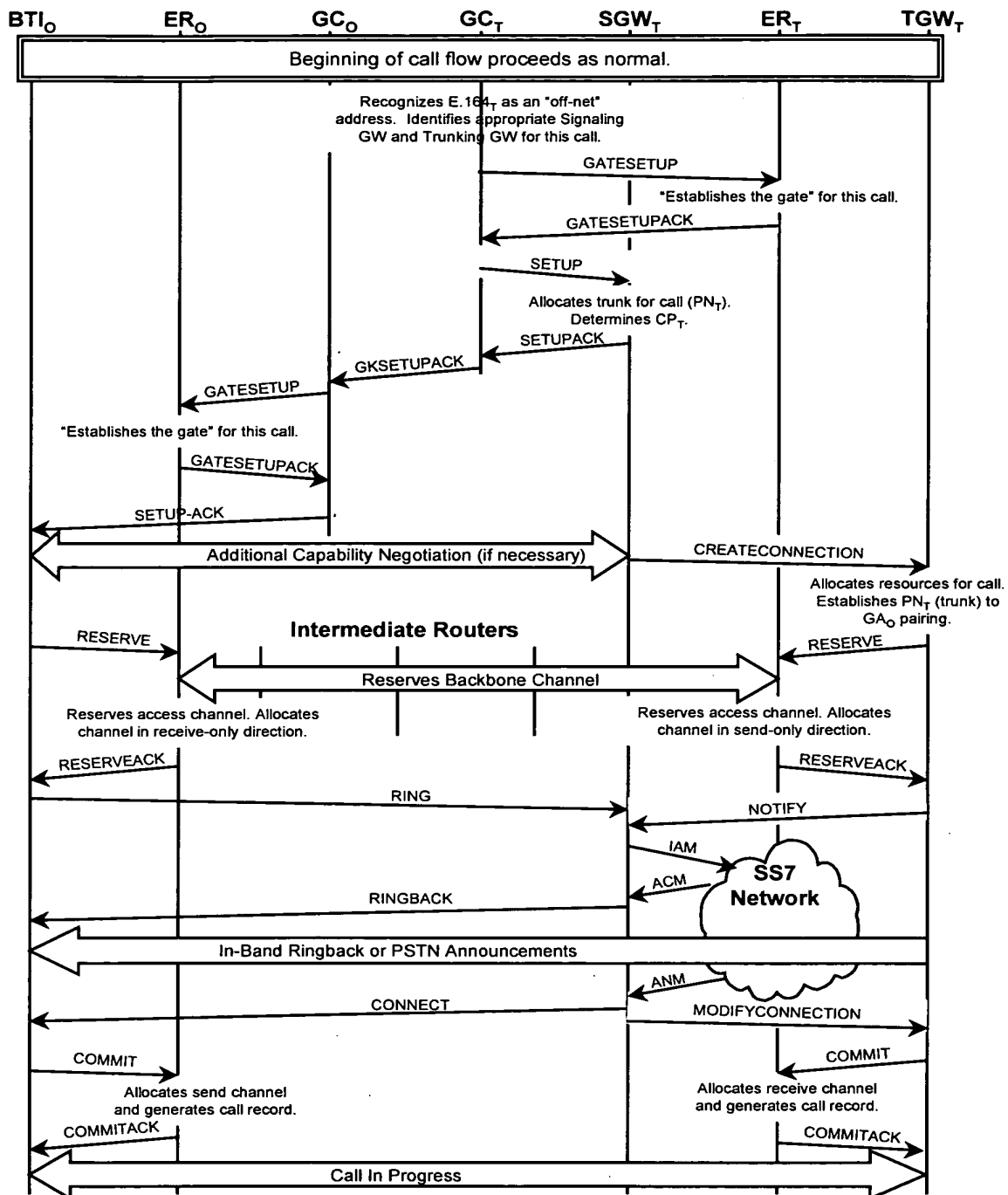


Figure 9

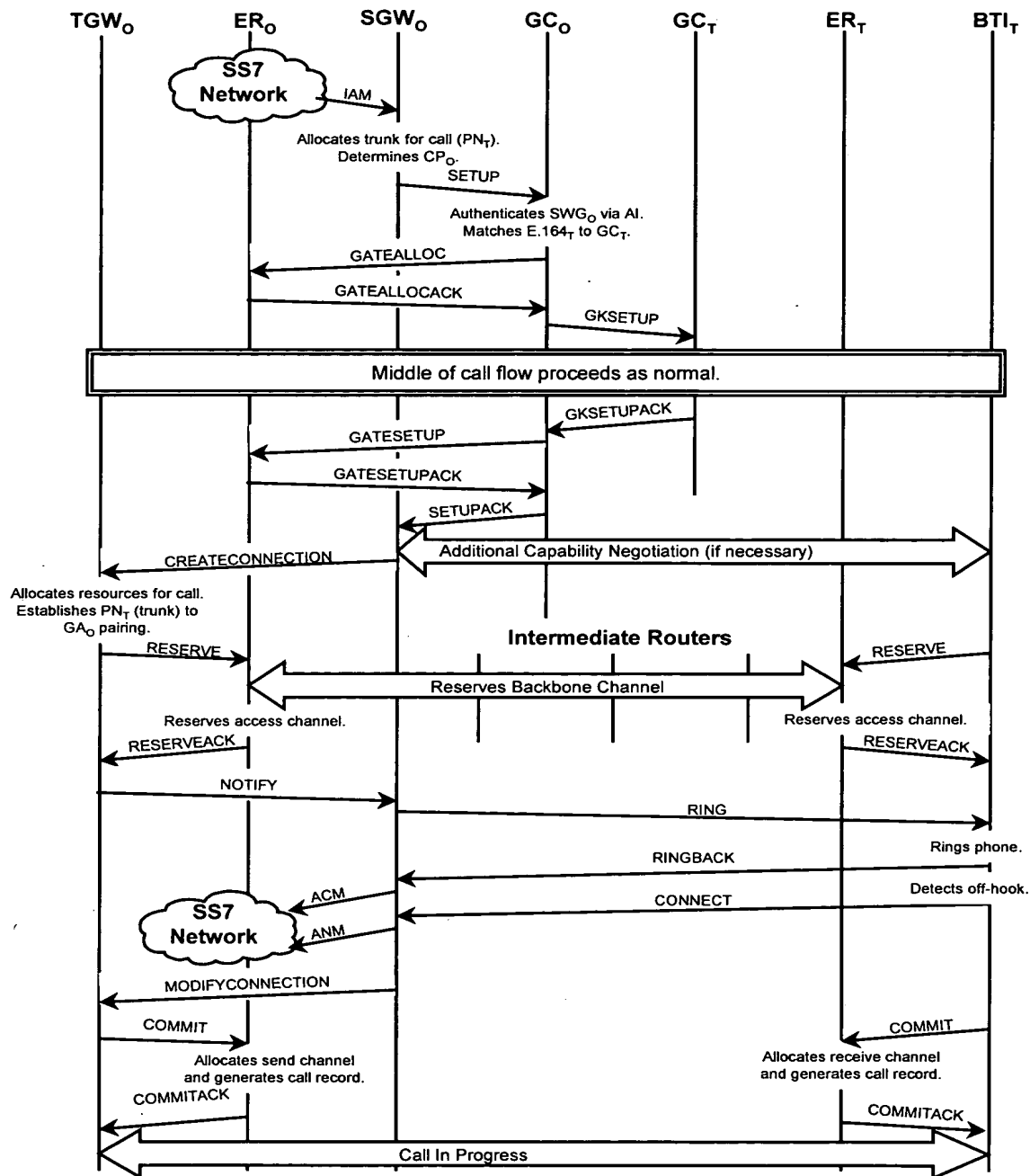


Figure 10

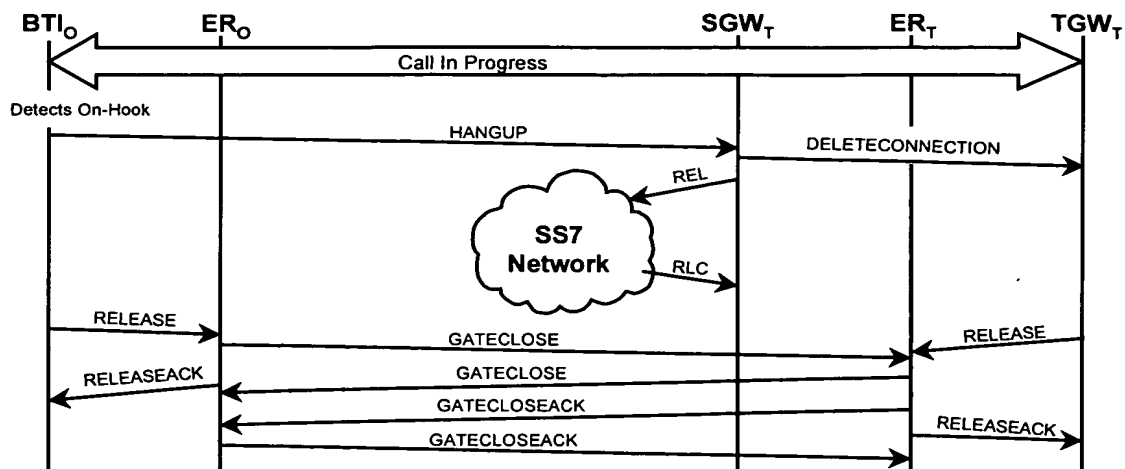


Figure 11

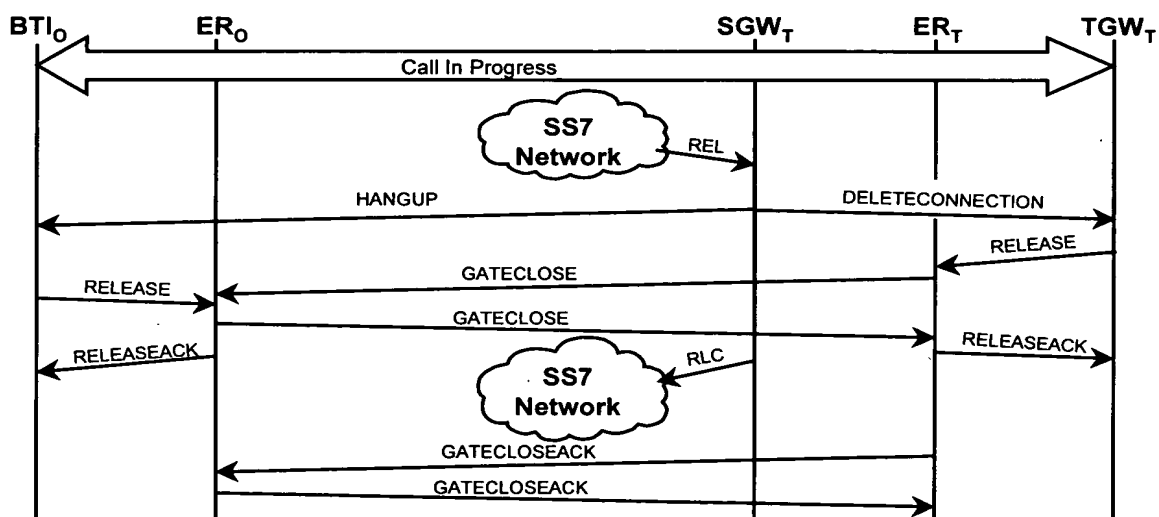


Figure 12

THE UNIVERSITY OF CHICAGO

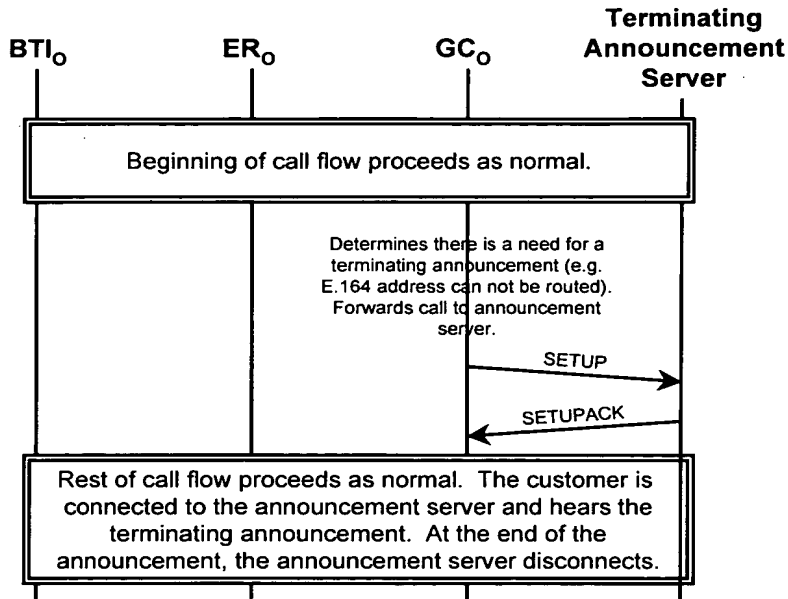


Figure 13

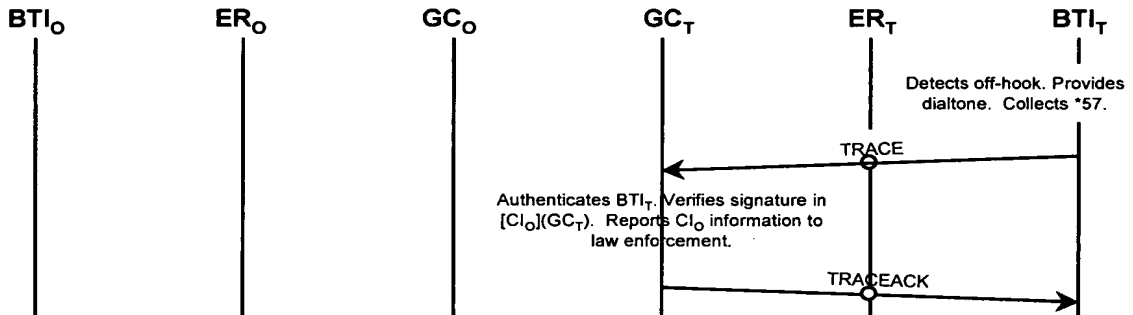


Figure 14

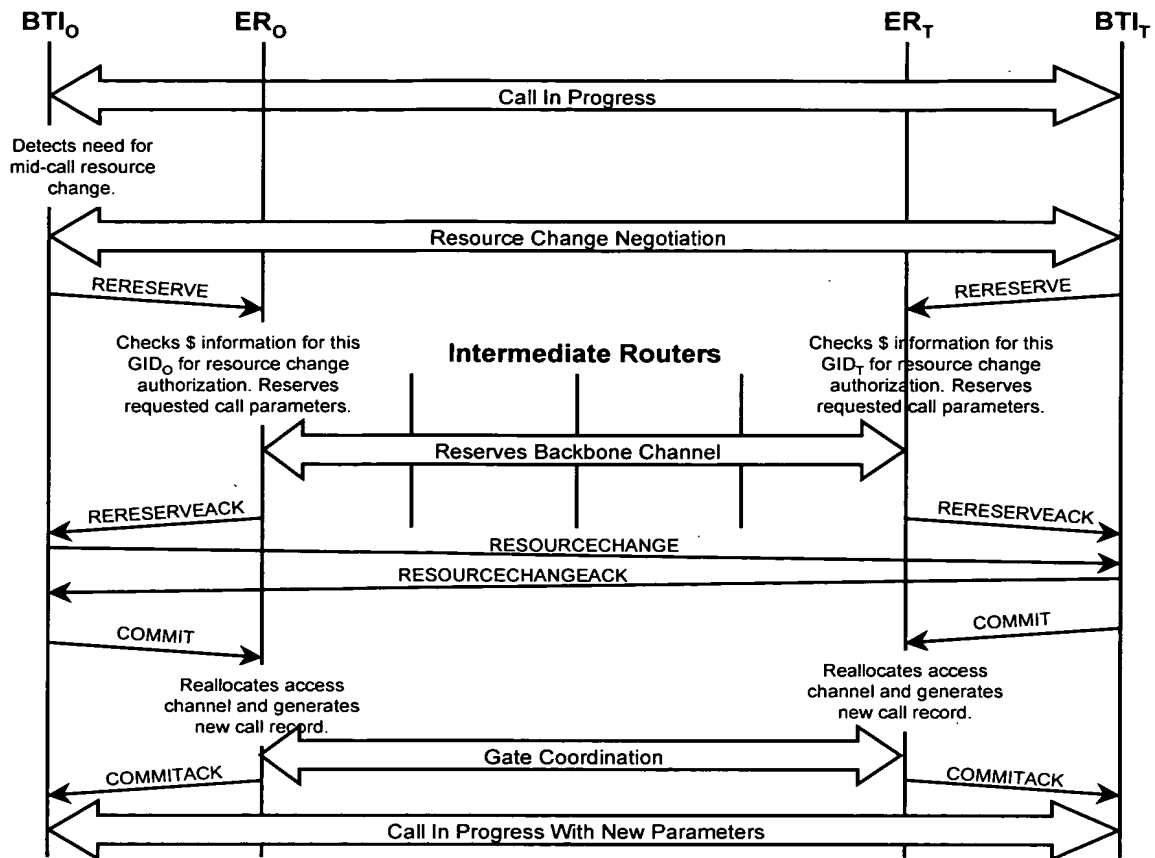


Figure 15

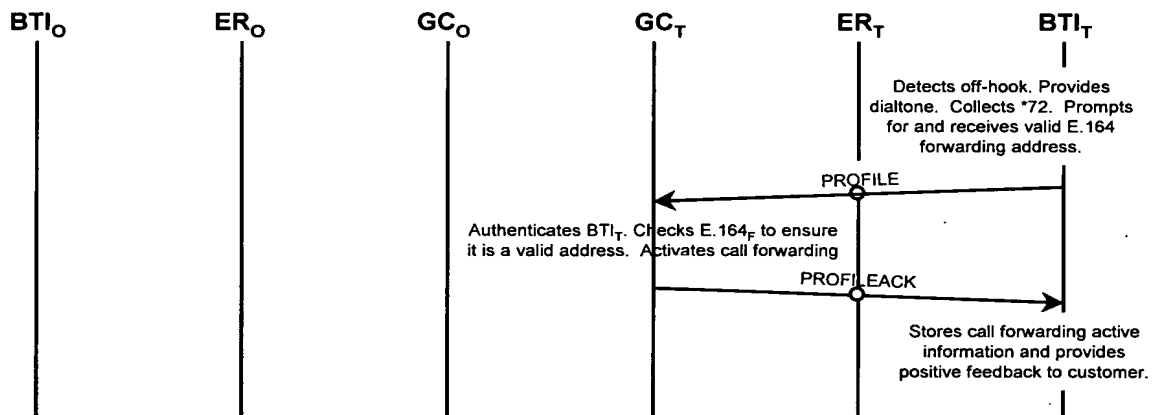


Figure 16

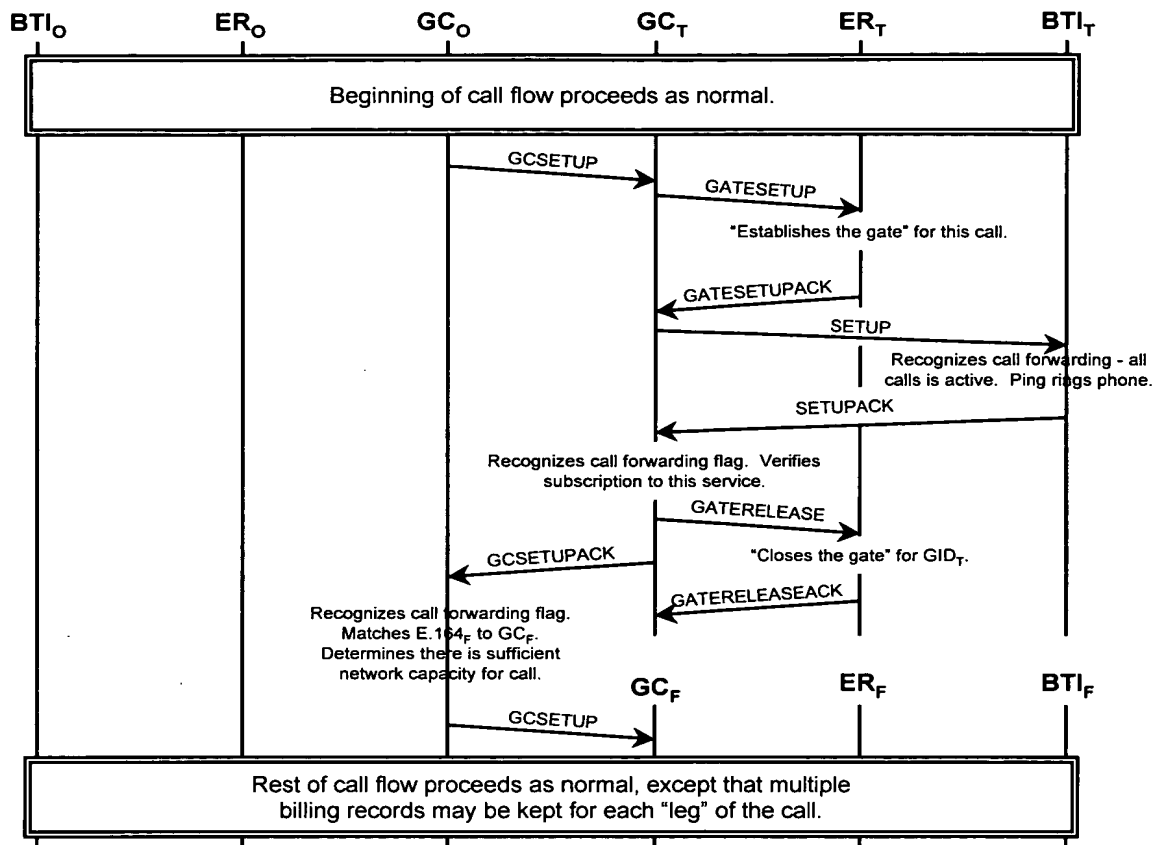


Figure 17

© 2000 "C" 2000 "C" 2000 "C"

```

sequenceDiagram
    participant BTI_O
    participant ER_O
    participant GC_O
    participant GC_T
    participant ER_T
    participant BTI_T
    participant GC_F
    participant ER_F
    participant BTI_F

    Note over BTI_O, ER_O, GC_O, GC_T, ER_T, BTI_T: Beginning of call flow proceeds as normal.

    GC_O->>GC_T: GCSETUP
    GC_T->>ER_T: GATESETUP
    Note over ER_T: "Establishes the gate" for this call.
    ER_T->>GC_T: GATESETUPACK
    ER_T->>BTI_T: SETUP
    Note over BTI_T: X
    Note over BTI_T: BTI_T out of service.
    GC_T->>ER_T: GATERELEASE
    Note over ER_T: "Closes the gate" for GID_T.
    ER_T->>GC_T: GATERELEASEACK
    GC_T->>GC_O: GCSETUPACK
    Note over GC_O: Recognizes call forwarding flag.  
Matches E.164_F to GC_F.  
Determines there is sufficient  
network capacity for call.
    GC_O->>GC_F: GCSETUP

    Note over BTI_O, ER_O, GC_O, GC_T, ER_T, BTI_T: Rest of call flow proceeds as normal, except that multiple  
billing records may be kept for each "leg" of the call.
  
```

Figure 18

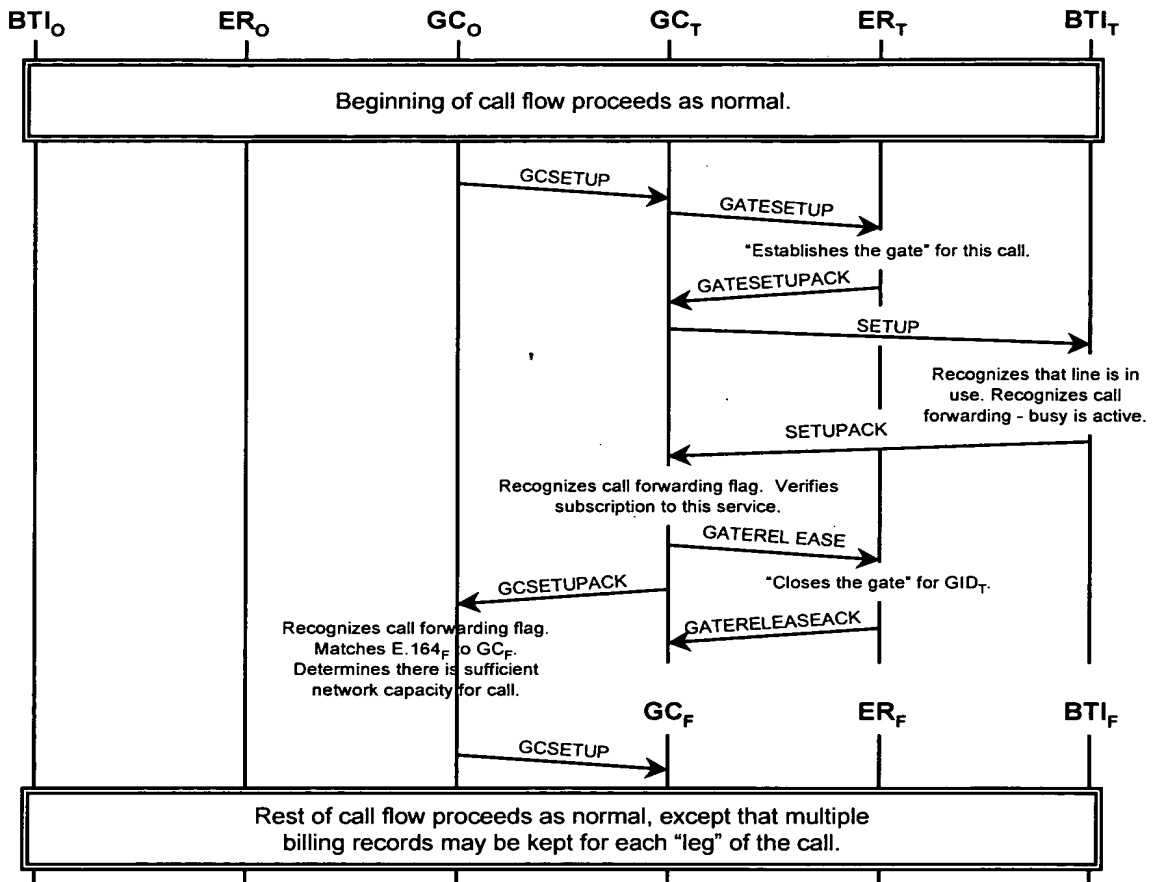


Figure 19

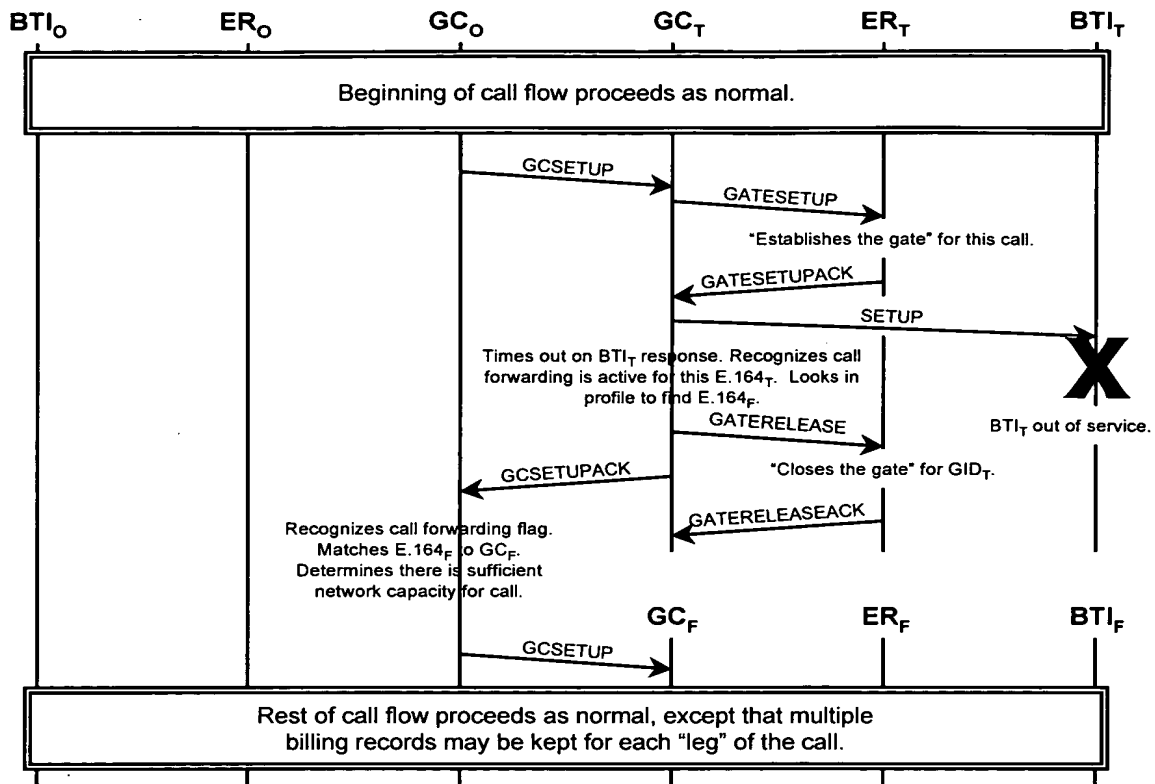


Figure 20

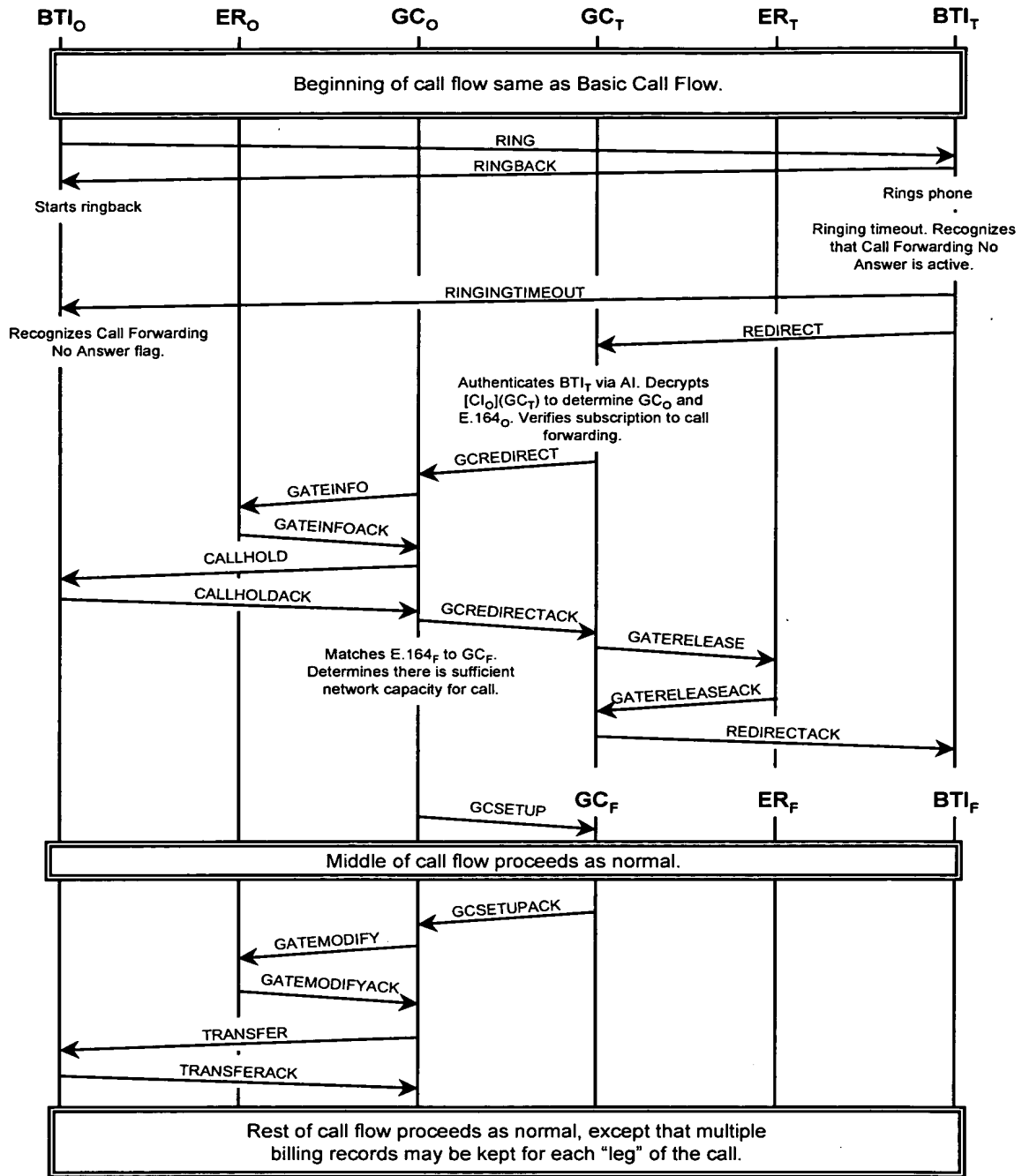


Figure 21

```

sequenceDiagram
    participant BTI_O
    participant ER_O
    participant GC_O
    participant GC_T
    participant ER_T
    participant BTI_T
    participant GC_F
    participant ER_F
    participant BTI_F

    Note over BTI_O, ER_O, GC_O, GC_T, ER_T, BTI_T: Beginning of call flow proceeds as normal.

    GC_O->>GC_T: GCSETUP
    GC_T->>ER_T: GATESETUP
    ER_T-->>GC_T: GATESETUPACK
    ER_T->>BTI_T: SETUP
    Note over BTI_T: BTI_T out of service. (X)
    GC_T->>ER_T: GATERELEASE
    ER_T-->>GC_T: GATERELEASEACK
    GC_T->>GC_O: GCSETUPACK
    Note over GC_O: Recognizes call forwarding flag.  
Matches E.164_F to GC_F.  
Determines there is sufficient  
network capacity for call.
    GC_O->>GC_F: GCSETUP

    Note over BTI_O, ER_O, GC_O, GC_T, ER_T, BTI_T: Rest of call flow proceeds as normal, except that multiple  
billing records may be kept for each "leg" of the call.
  
```

Figure 22

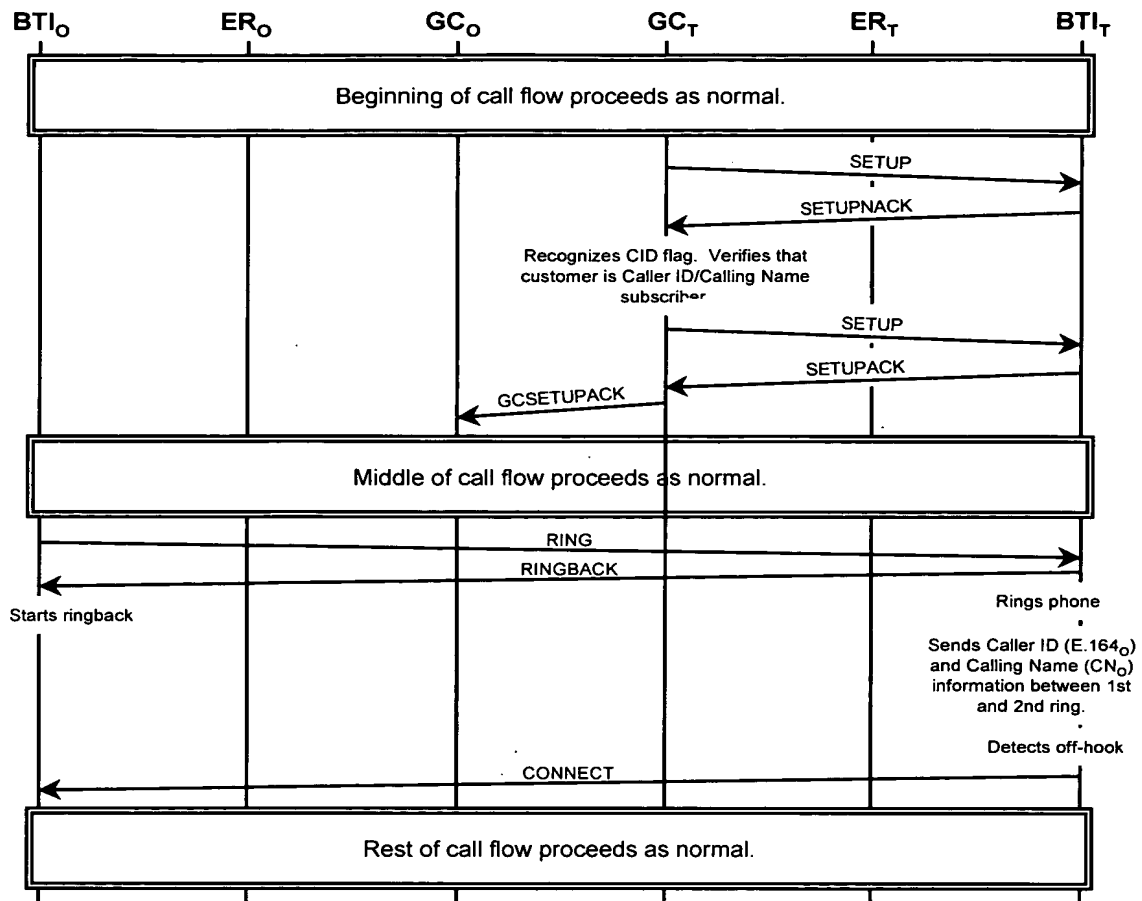


Figure 23

64000 64000 64000

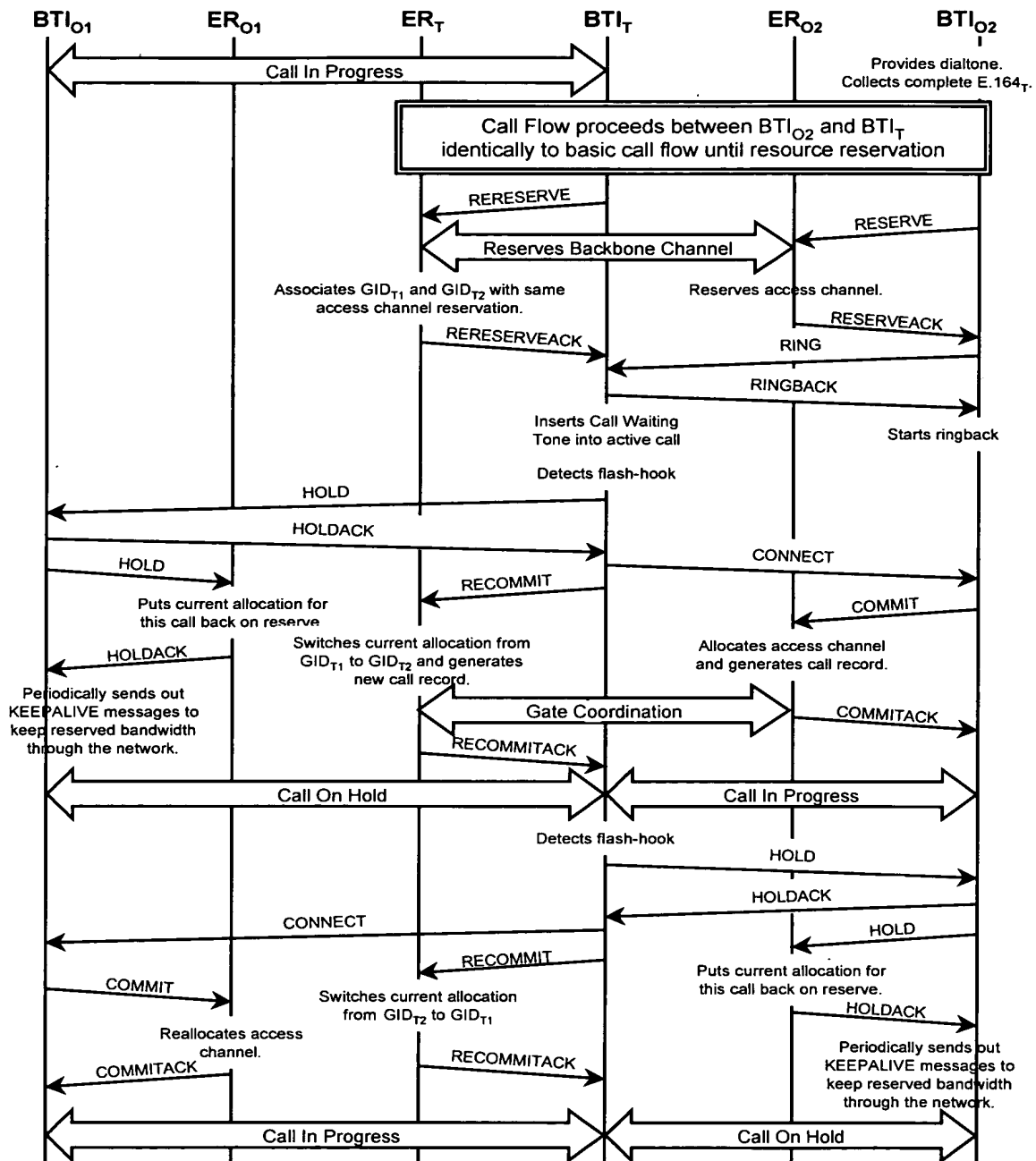


Figure 24

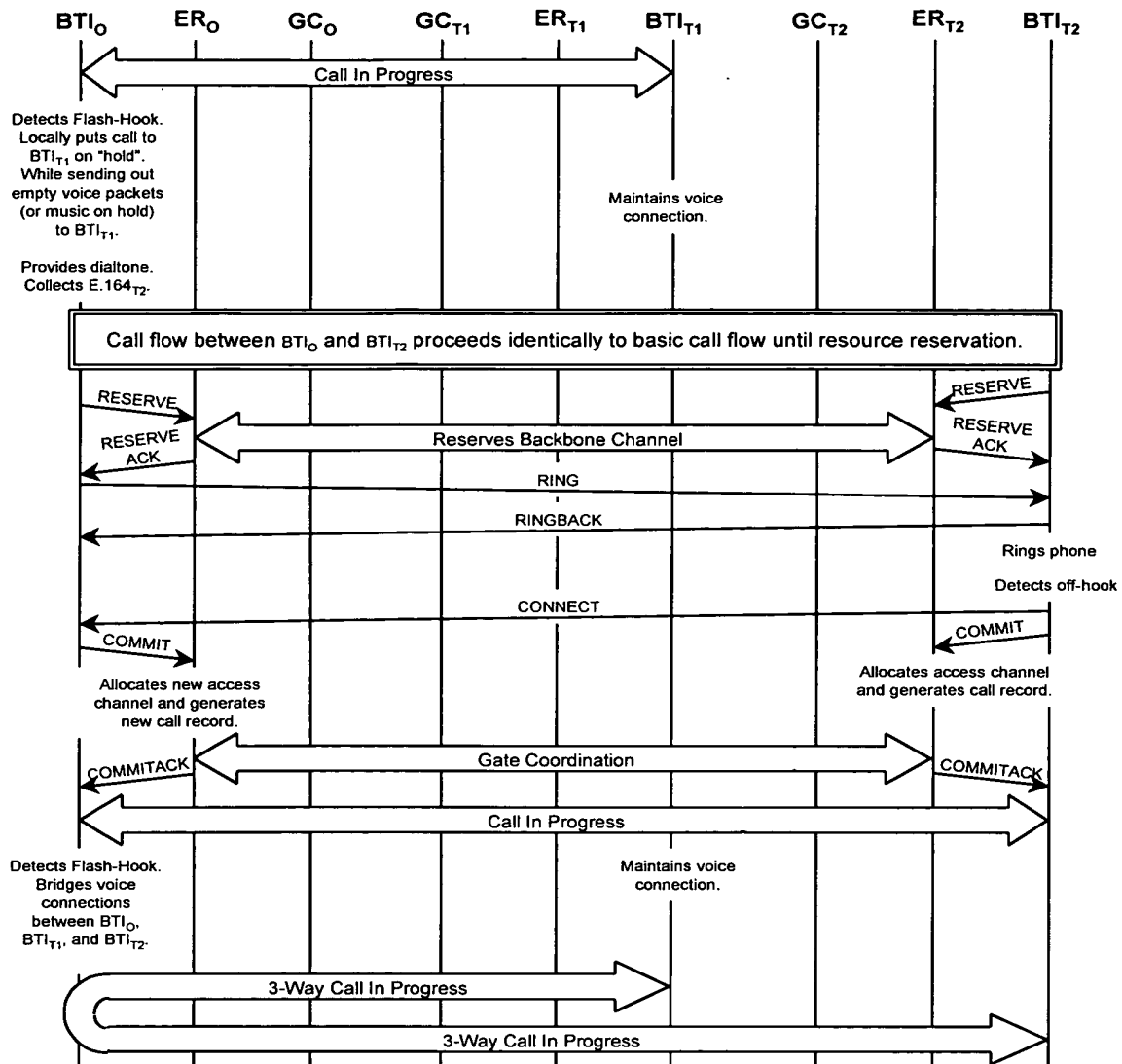


Figure 25

The diagram illustrates the sequence of events for a 3-Way Call In Progress. The participants involved are BTI_O, ER_O, GC_O, GC_{T1}, ER_{T1}, BTI_{T1}, Bridge, GC_{T2}, ER_{T2}, and BTI_{T2}.

Call In Progress: BTI_O detects a flash-hook and initiates a call held by GC_O. GC_O then initiates a call held by ER_{T1}, which connects to BTI_{T1}. BTI_{T1} connects to the Bridge, which connects to GC_{T2}, which connects to ER_{T2}, which connects to BTI_{T2}.

REDIRECT: GC_O sends a REDIRECT message to GC_{T1}. GC_{T1} sends a GATEINFO message to the Bridge, which responds with a GATEINFO ACK. The Bridge sends a SETUP message to GC_{T1}, which responds with a SETUPACK. The Bridge sends a GATE MODIFY message to GC_{T1}, which responds with a GATE MODIFYACK. The Bridge sends a TRANSFER message to GC_{T1}, which responds with a TRANSFERACK. The Bridge sends a RESERVE message to GC_{T1}, which responds with a RESERVE ACK. The Bridge sends a RING message to GC_{T1}, which responds with a CONNECT. The Bridge sends a COMMIT message to GC_{T1}, which responds with a COMMITACK. The Bridge sends a CALL message to GC_{T1}, which responds with a GCREDIRECT. The Bridge sends a REDIRECT message to GC_{T1}, which responds with a REDIRECTACK. The Bridge sends a GATE RELEASE message to GC_{T1}, which responds with a GATE RELEASE ACK. The Bridge sends a REDIRECT message to GC_{T1}, which responds with a REDIRECTACK. The Bridge sends a REDIRECT message to GC_{T1}, which responds with a REDIRECTACK.

3-Way Call In Progress: The call is now in progress between BTI_O and BTI_{T2} via the Bridge. The Bridge sends a REDIRECTACK message to GC_{T1}.

Figure 27

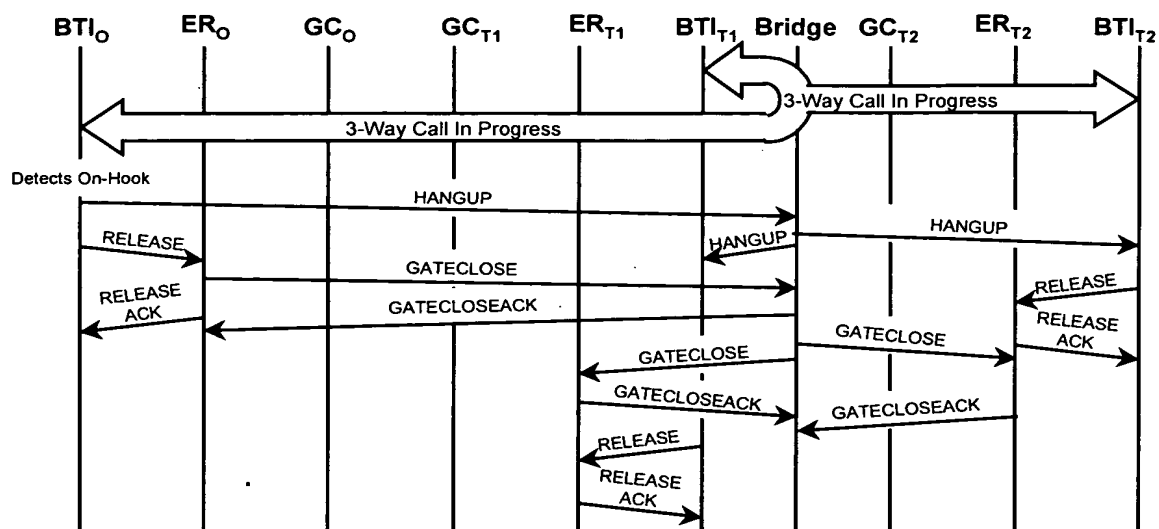


Figure 28

661030 " 2 293600

The diagram illustrates the sequence of events for a 3-way call setup and coordination between various entities. The entities are represented by vertical lanes: BTI_O, ER_O, GC_O, GC_{T1}, ER_{T1}, BTI_{T1}, Bridge, GC_{T2}, ER_{T2}, and BTI_{T2}.

3-Way Call In Progress: This phase is indicated by a large double-headed arrow at the top. It includes the following messages:

- ER_{T1} to BTI_{T1}: RELEASE
- BTI_{T1} to ER_{T1}: HANGUP
- ER_{T1} to BTI_{T1}: RELEASE ACK
- ER_{T1} to Bridge: GATECLOSE
- Bridge to ER_{T1}: GATECLOSEACK
- GC_{T1} to GC_O: SPLICE
- GC_O to ER_O: GATEINFO
- ER_O to GC_O: GATEINFO ACK
- GC_O to GC_{T2}: GCSPLICE
- GC_{T2} to ER_{T2}: GATEINFO
- ER_{T2} to GC_{T2}: GATEINFO ACK
- GC_{T2} to ER_{T2}: GATE MODIFY
- ER_{T2} to GC_{T2}: GATE MODIFY ACK
- GC_{T2} to BTI_{T2}: TRANSFER
- BTI_{T2} to ER_{T2}: RESERVE
- ER_{T2} to BTI_{T2}: RESERVE ACK
- BTI_{T2} to GC_{T2}: TRANSFERACK
- GC_{T2} to GC_O: GCSPLICEACK
- GC_O to ER_O: GATE MODIFY
- ER_O to GC_O: GATE MODIFYACK
- GC_O to BTI_O: TRANSFER
- BTI_O to ER_O: RESERVE
- ER_O to BTI_O: RESERVE ACK
- BTI_O to GC_O: TRANSFERACK
- GC_O to BTI_{T1}: SPLICEACK
- GC_{T1} to Bridge: RING
- Bridge to GC_{T1}: CONNECT
- GC_{T1} to BTI_{T1}: COMMIT
- BTI_{T1} to GC_{T1}: COMMITACK
- GC_{T1} to GC_{T2}: COMMIT
- GC_{T2} to GC_{T1}: COMMITACK

Gate Coordination: This phase is indicated by a large double-headed arrow at the bottom. It includes the following messages:

- GC_O to BTI_O: COMMIT
- BTI_O to GC_O: COMMITACK
- GC_O to GC_{T2}: COMMIT
- GC_{T2} to GC_O: COMMITACK

Call In Progress: This phase is indicated by a large double-headed arrow at the very bottom.

Figure 29

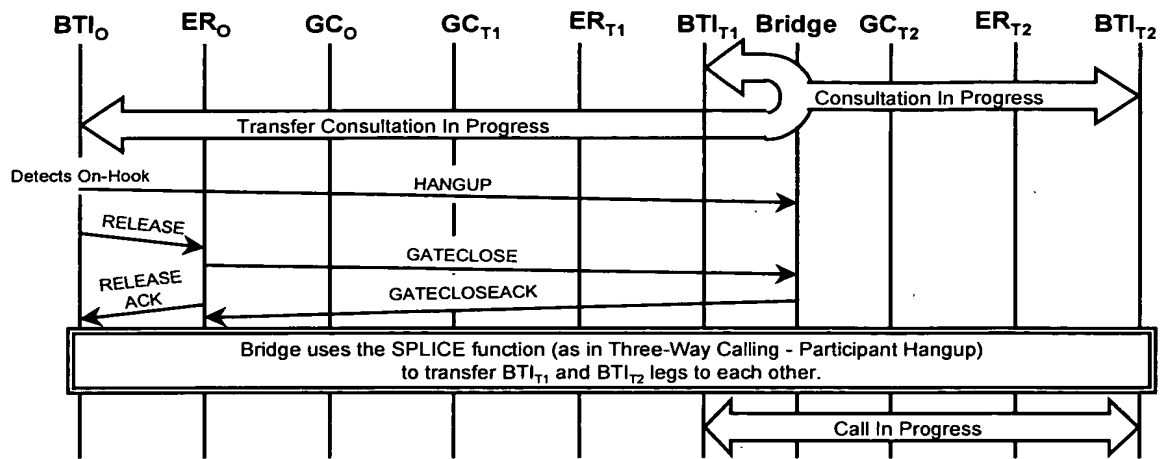


Figure 30

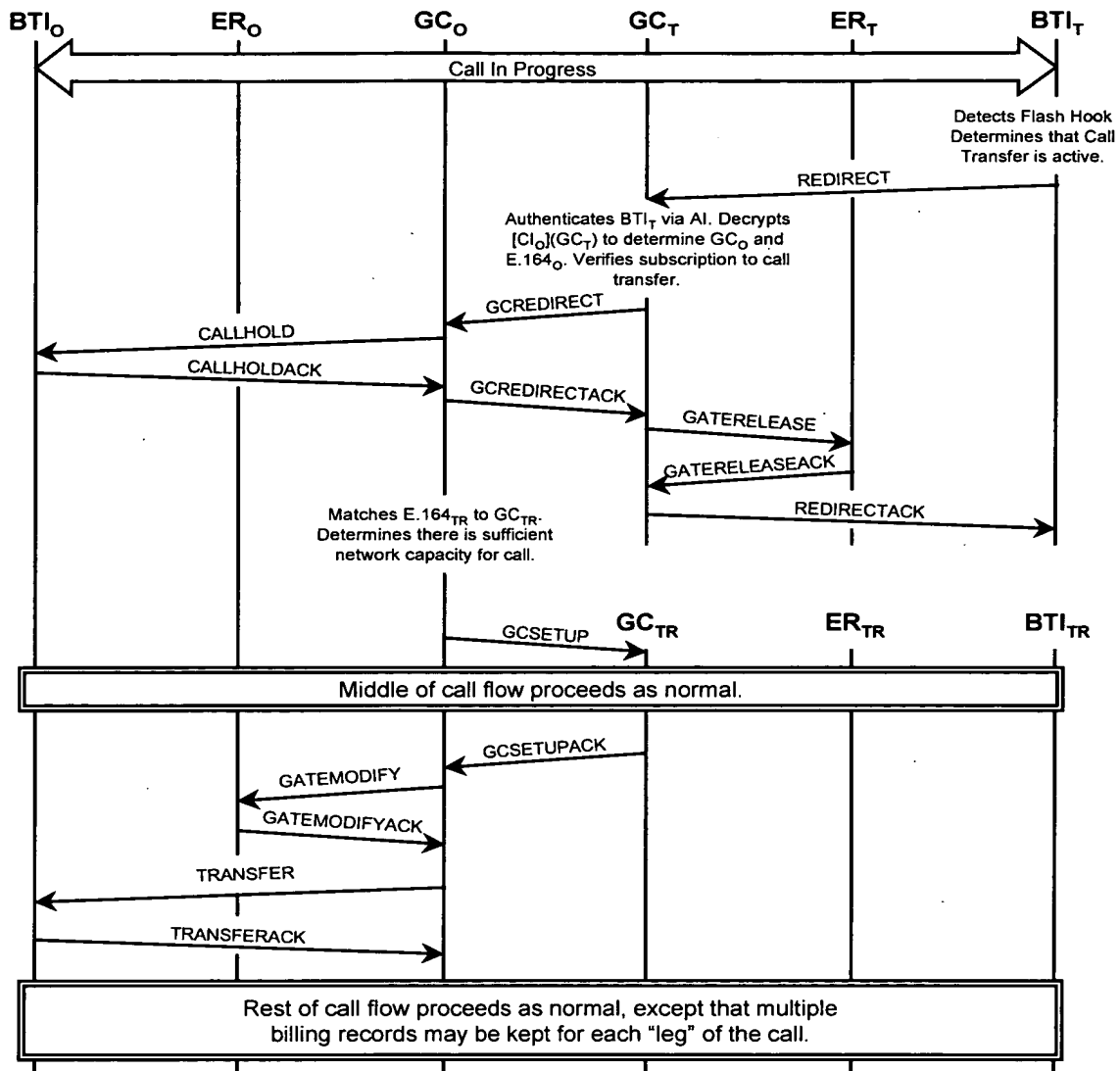


Figure 31

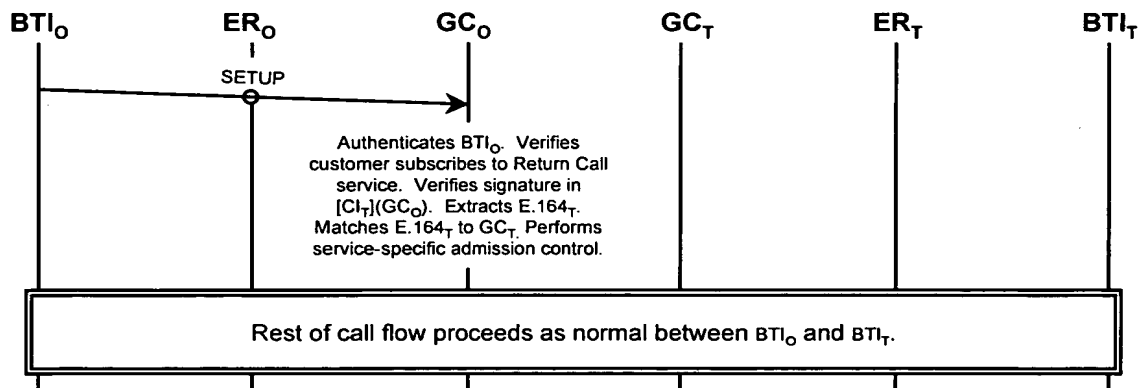


Figure 32